



(19)

(11) Publication number:

04369727 A

Generated Document.

PATENT ABSTRACTS OF JAPAN

(21) Application number: 03146143

(51) Intl. Cl.: G06F 9/32 G06F 9/26

(22) Application date: 18.06.91

(30) Priority:

(43) Date of application publication: 22.12.92

(84) Designated contracting states:

(71) FUJI ELECTRIC CO LTD
Applicant: FUJI FACOM CORP

(72) Inventor: HIROSE RYOICHI

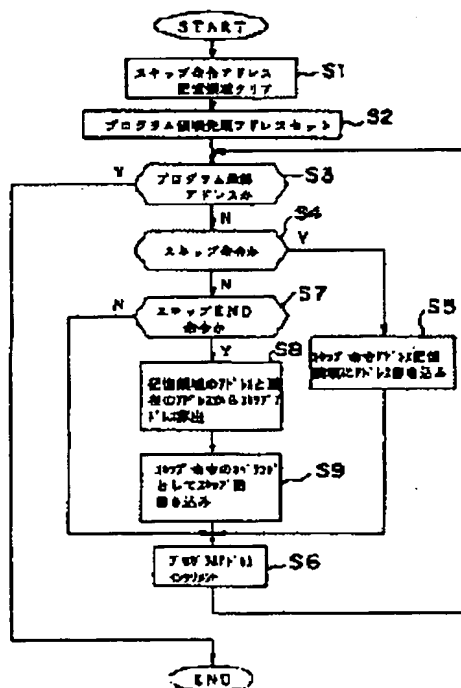
(74) Representative:

(54) ARITHMETIC UNIT

(57) Abstract:

PURPOSE: To shorten the processing time of a skip instruction and the like in an arithmetic unit having an instruction skipping the processing between the specified instructions based on a bit condition designated by means of the operand of the instruction.

CONSTITUTION: An area writing the number of steps with a skip instruction is provided for the operand of the skip instruction. A program which is prepared by a user is stored in the program of an arithmetic unit and the program is retrieved, whereby the number of the steps between the skip instruction and the skip END instruction is obtained (S4-S8). The number of the steps is written into the operand of the skip instruction (S9). When a bit condition is realized at the time of executing the program, the number of the steps written in the operand of the skip instruction is added to the value of a program counter and the address is increased (S6). Thus, an execution address can be advanced next to the skip END instruction.



COPYRIGHT: (C)1992 JPO&Japio

特開平4-369727

(43) 公開日 平成4年(1992)12月22日

(51) Int.Cl.⁵G 0 6 F 9/32
9/26

識別記号

3 4 0 B 9189-5B
3 2 0 F 9193-5B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数2(全5頁)

(21) 出願番号

特願平3-146143

(22) 出願日

平成3年(1991)6月18日

(71) 出願人 000005234

富士電機株式会社

神奈川県川崎市川崎区田辺新田1番1号

(71) 出願人 000237156

富士ファコム制御株式会社

東京都日野市富士町1番地

(72) 発明者 広瀬 良一

東京都日野市富士町1番地 富士ファコム
制御株式会社内

(74) 代理人 弁理士 大菅 義之

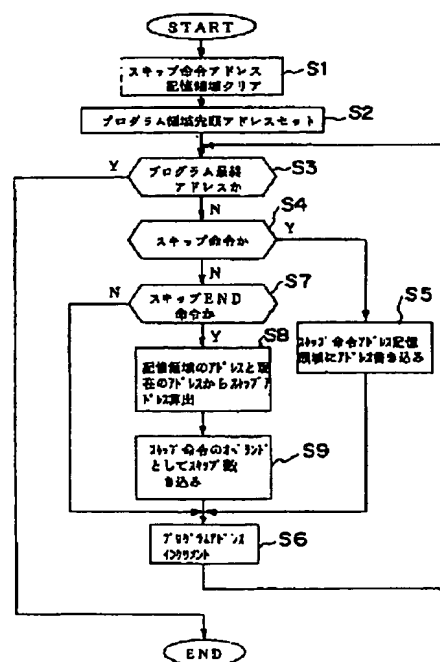
(54) 【発明の名称】 演算装置

(57) 【要約】

【目的】 本発明は、命令のオペランドで指定されるビット条件に基づいて特定の命令間の処理をスキップする命令を有する演算装置において、スキップ命令などの処理時間を短縮することを目的とする。

【構成】 スキップ命令のオペランドに、スキップEND命令との間のステップ数を書き込む領域を設けている。ユーザが作成したプログラムが演算装置のプログラム格納に格納された後、そのプログラムを検索してスキップ命令とスキップEND命令間のステップ数を求め(S4～S8)、そのステップ数をスキップ命令のオペランドに書き込む(S9)。プログラムの実行時にビット条件が成立した場合には、スキップ命令のオペランドに書き込まれているステップ数をプログラムカウンタの値に加算し、さらにそのアドレスをインクリメントすることで(S6)、実行アドレスをスキップEND命令の次に進めることができる。

実施例のフローチャート



【特許請求の範囲】

【請求項1】命令のオペランドで指定されるビット条件が成立したとき又は成立しなかったとき、特定の命令間の処理をスキップする命令を有する演算装置において、前記演算装置のプログラム格納領域に格納されたプログラムを検索して前記特定命令を検出する命令検出手段(1)と、この命令検出手段(1)で検出された特定命令間のステップ数を算出する算出手段(2)と、この算出手段(2)で算出されたステップ数を、前記ビット条件が設定されている命令のオペランドに設定する設定手段(3)とを備えることを特徴とする演算装置。

【請求項2】スキップ命令のオペランドで指定されるビット条件が成立したとき又は成立しなかったとき、実行アドレスをスキップEND命令の次のアドレスに変更する命令を有する演算装置において、前記命令検出手段(1)は、演算装置のプログラム格納領域を検索してスキップ命令とスキップEND命令とを検出し、前記算出手段(2)は、スキップ命令とスキップEND命令間のステップ数を算出し、前記設定手段(3)は、前記算出手段(2)で算出されたステップ数を、スキップ命令のオペランドに設定し、プログラム実行時にスキップ命令のビット条件が成立したとき又は成立しなかったときには、スキップ命令のオペランドで指定されるステップ数をプログラムカウンタに加算した値を実行アドレスとすることを特徴とする請求項1記載の演算装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、命令のオペランドで指定されるビット条件に基づいて特定の命令間の処理をスキップする命令を有する演算装置に関する。

【0002】

【従来の技術】従来、スキップ命令とスキップEND命令とを有し、スキップ命令のオペランドで指定される特定のビット条件が成立しないときには、次のアドレスの命令を実行し、ビット条件が成立したときには、スキップEND命令の次のアドレスの命令を実行する演算装置がある。

【0003】この種の演算装置では、プログラムの実行時に、スキップ命令のビット条件が成立した場合には、プログラムの格納領域にあるスキップ命令以降の命令を1命令づつ検索し、検索した命令がスキップEND命令でなければ何も処理を実行せず次の命令を検索し、スキップEND命令を検出したなら、その次の命令から処理を実行するようになっている。

【0004】

【発明が解決しようとする課題】上述した従来の演算装置では、スキップ命令のオペランドで指定されるビット条件が成立し、スキップEND命令に進む場合に、プログラム格納領域にあるスキップ命令以降の命令を1命令づつ検索していくので、処理に時間がかかるという問題

点があった。

【0005】本発明の目的は、スキップ命令などの処理時間を短縮することである。

【0006】

【課題を解決するための手段】図1は、本発明の原理説明図である。オペランドで指定されるビット条件が成立したとき又は成立しないとき、特定の命令間の処理をスキップする命令を有する演算装置において、命令検出手段1は、格納されたプログラムを検索して特定命令を検出する。

【0007】算出手段2は、上記命令検出手段2で検出された特定命令間のステップ数を算出する。設定手段3は、算出手段2で算出されたステップ数を、ビット条件が設定されている命令のオペランドに設定する。

【0008】これにより、例えばスキップ命令とスキップEND命令とを有する演算装置では、スキップ命令のオペランドにスキップEND命令までのステップ数が書き込まれる。

【0009】

【作用】上記のようにして演算装置のプログラム格納領域に格納されているプログラムを実行する場合に、例えばスキップ命令の実行時にビット条件が成立したときには、プログラムカウンタにスキップ命令のオペランドで指定されるステップ数が加算される。

【0010】これにより、スキップ命令の次のアドレスとして直接スキップEND命令のアドレスが指定されるので、従来のようにスキップ命令とスキップEND命令との間の命令を1命令づつ取り出してスキップEND命令を検索する必要がなくなり、処理時間を大幅に短縮することができる。

【0011】

【実施例】以下、本発明の実施例を図面を参照しながら説明する。この実施例は、各種の電子機器のシーケンシャル制御に利用されるプログラマブルコントローラに関するものである。

【0012】図2は、プログラマブルコントローラ上でのユーザプログラムの表示の一例を示す図である。同図において、B0、B1等は、1/O装置のシンボルを示している。

【0013】ここで、スキップ命令(SKIP)とスキップEND命令(SEND)とを含むプログラムを、プログラマブルコントローラのプログラム格納領域に格納する場合の処理を説明する。

【0014】図3は、プログラマブルコントローラのプログラム格納領域に格納される制御内部命令を示しており、同図(a)は従来の制御内部命令、同図(b)は本発明の制御内部命令を示している。

【0015】従来は、スキップ命令は、1つのオペランドしか持っておらず、そのオペランドには、ユーザが設定するビット条件が記憶される。これに対して、実施例

3

の演算装置では、スキップ命令は2つのオペランドを有しており、ビット条件と、スキップ命令とスキップEND命令との間の相対アドレス（ステップ数）が記憶される。なお、ビット条件はユーザが設定するが、ステップ数は、後述する処理により演算装置内部で設定される。

【0016】ここで、スキップ命令のオペランドにステップ数を書き込む処理について図4のフローチャートを参照して説明する。ユーザが作成したプログラムをプログラム格納領域に格納する段階では、スキップEND命令がどのアドレスに割り付けられるか分からないので、スキップ命令には、スキップEND命令との相対ステップ数を格納する領域だけが設けられ、実際のステップ数は書き込まれない。

【0017】この状態で、まず、スキップ命令の格納アドレスを記憶するスキップ命令アドレス記憶領域をクリアする（図4、S1）。次に、プログラム格納領域の先頭アドレスをプログラムカウンタにセットする（S2）。そして、プログラムカウンタの値が最終アドレスに達したか否かを判別する（S3）。

【0018】最終アドレスでなければ、現在のアドレスの命令を解釈してスキップ命令か否かを判別する（S4）。スキップ命令であれば、ステップ5に進み現在のアドレスをスキップ命令アドレス記憶領域に書き込む。その後、ステップ6に進みプログラムカウンタの値をインクリメントしてステップ3に戻り上述した処理を繰り返す。

【0019】また、ステップ4の判別で解釈した命令がスキップ命令でなければ、次のステップ7で解釈した命令がスキップEND命令か否かを判別する。スキップEND命令でなければ、ステップ6に進みプログラムカウンタをインクリメントしてステップ3に戻り上述した処理を繰り返す。

【0020】一方、解釈した命令がスキップEND命令であったときには、スキップ命令アドレス記憶領域に記憶されているアドレスと現在のアドレスとから両命令間のステップ数を算出する（S8）。さらに、その算出したステップ数（スキップ数）をスキップ命令のオペランドに書き込む（S9）。

4

【0021】これらの処理によりプログラム格納領域のスキップ命令のオペランドには、ビット条件に続いてスキップEND命令との間のステップ数が設定される。そして、プログラムの実行時には、検出された命令がスキップ命令でかつ、ビット条件が成立している場合には、そのオペランドで指定されるステップ数がプログラムカウンタに加算される。さらに、そのプログラムカウンタの値がインクリメントされて、スキップEND命令の次のアドレスがネクストアドレスとして指示される。すなわち、スキップ命令のビット条件が成立した場合には、アドレスを直接スキップEND命令の次のアドレスに進めることができる。

【0022】従って、従来のようにスキップEND命令を検出するまで、命令を1命令ずつ検索する必要がないので、スキップ命令の処理時間を大幅に短縮できる。また、この場合、スキップ先アドレスは、演算装置内部で自動的に算出され、条件付分岐命令などのようにラベル等によりユーザが設定するアドレスではないので、アドレス変換のテーブル等を設ける必要がない。

【0023】

【発明の効果】本発明によれば、ユーザが作成したプログラムが演算装置のプログラム格納領域に格納された後、例えばスキップ命令のオペランドにスキップ先アドレスまでのステップ数が書き込まれる。これにより、プログラムの実行時にビット条件が成立したときには、スキップ命令の次のアドレスとして、直接スキップEND命令の次のアドレスを指定することができ、命令の処理時間を大幅に短縮することができる。

【図面の簡単な説明】

【図1】本発明の原理説明図である。

【図2】実施例のプログラムの一例を示す図である。

【図3】制御内部命令の説明図である。

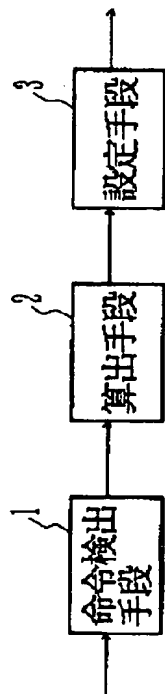
【図4】実施例のフローチャートである。

【符号の説明】

- 1 命令検出手段
- 2 算出手段
- 3 設定手段

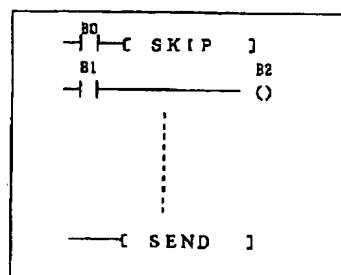
【図1】

本発明の原理説明図



【図2】

プログラムの一例を示す図



【図3】

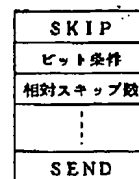
制御内部命令の説明図

制御内部命令（従来）



(a)

制御内部命令（本発明）



(b)

【図4】

実施例のフローチャート

